<u>AMENDMENTS TO THE CLAIMS</u>

<u>WE CLAIM:</u>

1. (Original) In a data storage system that includes a host computer coupled to a disk array controller that is in turn coupled to a plurality of data storage devices, a method providing improved data consistency checking and correction of user data stored in one or more data sectors on a data stripe, each data sector including a respective data header, the data stripe including a parity sector that in turn includes a parity header, the data stripe being distributed across the data storage devices, the method comprising:

for each data header, storing a respective first code into the data header that represents the user data stored in the data header's respective data segment;

storing a second code into the parity header, the second code representing a set of parity data stored in the parity sector; and,

storing a third code into the parity header, the third code representing a value obtained by encoding the first code and the second code, the third code being used to check the consistency of the user data and the parity data without performing respective operations on each byte of data in both the data sectors and the parity sector.

2. (Original) The method of claim 1, wherein:

(a) in the first code storing, each respective first code is a data check code sub-data sector (DCC_{ds}) code;

(b) in the second code storing, the second code is a data check code sub-parity sector (DCC_{ps}) code; and,

(c) in the third code storing, the third code is a data check code sub-sector (DCC_{ss}).

3. (Original) The method of claim 1, further comprising:

determining if the user data is corrupt, the determining comprising steps of:

reading the parity data and the user data;

for each respective data segment of the data segments:

(a) providing a calculated first code based on the user data stored in the respective data segment;

(b) comparing the calculated first code to the first code to determine if the stored user data is corrupt;

(c) if the user data is corrupt, regenerating the corrupt user data using certain other data sectors of the data sectors and the parity data; and,

(d) storing the regenerated user data in the respective data segment, the regenerated user data replacing the corrupt user data.

4. (Original) The method of claim 3, further comprising:

for each respective data sector of the data sectors:

(a) if the user data in the respective data sector is corrupt, generating a new first code from the regenerated user data; and,

(b) storing the new first code to the respective data sector's respective data header, the new first code replacing the first code.

5. (Original) The method of claim 1, further comprising:

determining if the parity data is corrupt, the determining comprising:

(a) providing a calculated DCCps based on the parity data;

(b) comparing the calculated DCCps to the DCCps to determine if the parity data is corrupt; and,

(c) if the parity data is corrupt, regenerating the corrupt parity data from the user data; and,

(d) storing the regenerated parity data to the parity sector, the regenerated parity data replacing the corrupt parity data.

6. (Original) The method of claim 5, further comprising:

if the parity data is corrupt, generating a new DCCps from the regenerated parity data; and,

storing the new DCCps to the parity header, the new DCCps replacing the DCCps.

7. (Currently Amended) The method of Claim 3 further comprising:

generating a new DCCss from the new DCC ps and each respective first code; and;

storing the new ~~first code~~ DCCss in the parity header, the new DCCss replacing the DCCss.

8. (Currently Amended) The method of Claim 4 further comprising:

generating a new DCCss from the new DCCps and each respective first code; and;

storing the new ~~first code~~ DCCss in the parity header, the new DCCss replacing the DCCss.

9. (Currently Amended) The method of claim 1, further comprising:

calculating a new ~~first code~~ DCCss from each respective first code and the second code; and, comparing the new ~~first code~~ DCCss to the ~~first code~~ DCCss to determine if there is an inconsistency between the user data in the parity data.

10. (Original) The method of claim 7, further comprising, if the new first code is not equal to the first code, reporting the inconsistency in the data stripe to a system administrator.

11. (Original) The method of claim 1, further comprising:

before reading, locking after the step of storing and before the step of performing, by the disk array controller, the data stripe to prevent a different entity as compared to the disk array controller from performing input/output operations on the data stripe; and,

after providing a data check code sub-sector, unlocking after performing the data stripe to allow the different entity to perform input/output to the data stripe.

12. (Original) A computer program product for use in conjunction with a processor, the computer program product comprising a first computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising a data consistency checking procedure to determine inconsistencies between parity data and user data stored in a stripe of data distributed across a plurality of data storage devices in a data storage system, the parity data being stored in a parity sector, the parity sector including a parity header, the user data being stored in one or more data sectors, each respective data sector of the data sectors including a data header, the data consistency checking procedure including instructions for:

for each data header, storing a respective first code into the data header that represents the user data stored in the data header's respective data segment;

caching a second code into the parity header, the second code representing a set of parity data stored in the parity sector; and,

putting a third code into the parity header, the third code representing a value obtained by encoding the first code and the second code, the third code being used to check the consistency of the user data and the parity data without performing respective operations on each byte of data in both the data sectors and the parity sector.

13. (Original) The computer program product of claim 12, wherein the data consistency checking procedure further including instructions for: processing said first code and said second code to check stripes.

14. (Original) An apparatus for performing data consistency checking operations, the apparatus comprising:

a first communications link coupling the apparatus to a computer;

a second communications link coupling the apparatus to a plurality of data storage devices;

a processor for fetching data and executing computer program instructions stored in a memory, the computer program instructions comprising a data consistency checking procedure for:

storing a respective first code of one or more first codes into a set of data headers in one or more data segments, the data segments being located in a data stripe distributed across the plurality of data storage devices, the first code representing a set of user data stored in the data header's respective data segment;

providing a parity header with a second code to represent a set of parity data stored in a parity sector in the data stripe, the parity header being part of the parity sector; and

providing the parity header with a third code by encoding each respective first code with the second code; such that the third code can be used to check the consistency of a set of user data and a set of parity data without the need to respective operations on each byte of data in both the data sectors and the parity sector, the user data being stored in the one or more data sectors, the parity data being stored in the parity sector.

15. (Currently Amended) A system for performing data consistency checking, the system comprising:

a computer for sending a data consistency check request to a controller, the computer being coupled to the controller, the controller being responsive to the receipt of the request, the request pertaining to checking for differences between a set of parity data and a set of user data stored in a data stripe distributed across a plurality of data storage devices coupled to the controller;

a procedure stored in a memory coupled to the controller, the procedure for:

storing a respective first code of one or more first codes into a set of data headers in one or more data segments, the data segments being part of the data stripe, the first code representing at least a subset of the user data stored in the data header's respective data segment;

caching a second code into a parity header, the first code representing a set of parity data stored in a parity sector in the data stripe, the parity header being part of the parity sector; and

putting a third code into the parity header, the third code being obtained by encoding the first code with the second code, the third code being used to check the consistency of a set of user data and a set of parity data without the need to respective operations on each byte of data in both the data sectors and the parity sector, the user data being stored in the one or more data sectors, the parity data being stored in the parity sector.

16. (Original) In a data storage system that includes a host computer coupled to a disk array controller that is in turn coupled to a plurality of data storage devices, a method providing a user with a number of different levels of data consistency checking techniques, the data checking techniques to be performed on user data stored in one or more data sectors on a data stripe, each data sector including a respective data header, the data stripe including a parity sector that in turn includes a parity header, the data stripe being distributed across the data storage devices, the method comprising:

providing a number of different types of data consistency checking levels to a user;

receiving an indication of a selected type of the number of different types of data consistency checking levels; and,

performing a particular data consistency checking technique between the parity data and the user data based on the selected type.

17. (Original) The method of claim 16, wherein said performing is only performed if there is user data stored in the data stripe and if the stored user data has been modified since a last data consistency check operation has been performed between the user data and the parity data.

18. (Original) The method of claim 16, wherein the particular data consistency checking technique comprises:

reading data in the data stripe;

generating a calculated third code from a stored second code in the parity header and from a set of stored first code values, a respective first code value of the set of first code values being stored in each respective data header of the data stripe; and,

comparing the calculated third code to a stored third code in the parity header, such that the comparison identifies whether there the user data is inconsistent with the parity data without the need to perform respective operations on each byte of data in both the data sectors and the parity sector.

19. (Original) The method of claim 17, wherein the particular data consistency checking technique comprises:

reading data in the data stripe;

generating a calculated third code from a stored second code in the parity header and from a set of stored first code values, a respective first code value of the set of first code values being stored in each respective data header of the data stripe; and,

comparing the calculated third code to a stored third code in the parity header, such that the comparison identifies whether there the user data is inconsistent with the parity data without the need to perform respective operations on each byte of data in both the data sectors and the parity sector.

20. (Original) The method of claim 15, wherein the particular data consistency checking technique comprises:

reading data in the data stripe; and,

performing operations on each byte of data both in the parity sector and each respective data sector to determine if there is a difference between the parity data and the user data.

21. (Original) A computer program product for use in conjunction with a processor, the computer program product comprising a first computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising a data consistency checking procedure to determine inconsistencies between parity data and user data stored in a stripe of data distributed across a plurality of data storage

devices in a data storage system, the parity data being stored in a parity sector, the parity sector including a parity header, the user data being stored in one or more data sectors, each respective data sector of the data sectors including a data header, the data consistency checking procedure including instructions for:

providing a number of different types of data consistency checking levels to a user;

receiving an indication from the user of a selected type of the number of different types of data consistency checking levels; and,

performing a particular data consistency checking technique between the parity data and the user data based on the selected type.

22. (Original) An apparatus for providing a user with a number of different levels of data consistency checking techniques, the apparatus comprising:

a first communications channel for coupling the apparatus to a computer;

a second communications channel coupling the apparatus to a plurality of data storage devices;

a processor for fetching data and executing computer program instructions stored in a memory, the computer program instructions comprising a data consistency checking procedure for:

providing a number of different types of data consistency checking levels to a user;

receiving an indication of a selected type of the number of different types of data consistency checking levels; and,

performing a particular data consistency checking technique between the parity data and the user data based on the selected type.

23. (Original) A method for checking data consistency of data stored in one or more data sectors of a data stripe, each data sector including a respective data header, the data stripe including a parity sector that includes a parity header, the method comprising:

for each data header, storing a respective first code in the data header that represents the user data stored in the data header's respective data segment;

storing a second code into a parity header, the second code representing a set of parity data stored in a parity sector; and, storing a third code into the parity header, the third code representing a value obtained by encoding the first code and the second code, the third code being used to check the consistency of the user data and the parity data.

24. (Original) The method of claim 23, wherein said consistency checking is performed without performing respective operations on each byte of data in both the data sectors and the parity sector.

25. (Original) The method of claim 23, wherein:

(a) each respective first code comprises a data check code sub-data sector (DCC_{ds}) code;

(b) the second code comprises a data check code sub-parity sector (DCC_{ps}) code; and,

(c) the third code comprises a data check code sub-sector (DCC_{ss}).

26. (Original) The method of claim 23, further comprising determining if the user data is corrupt, wherein the determining comprising:

reading the parity data and the user data; and

for each respective data segment:

(a) providing a calculated first code based on the user data stored in the respective data segment;

(b) comparing the calculated first code to the first code to determine if the stored user data is corrupt;

(c) if the user data is corrupt, regenerating the corrupt user data using certain other data sectors of the data sectors and the parity data; and,

(d) storing the regenerated user data in the respective data segment, the regenerated user data replacing the corrupt user data.

27. (Original) The method of claim 26, further comprising:

for each respective data sector:

(a) if the user data in the respective data sector is corrupt, generating a new first code from the regenerated user data; and,

(b) storing the new first code to the respective data sector's respective data header, the new first code replacing the first code.

28. (Original) A method for checking data consistency of data stored in one or more data sectors of a data stripe, each data sector including a respective data header, the data stripe including a parity sector that includes a parity header, the method comprising:

for each data header, storing a respective first code comprises a data check code sub-data sector code in the data header that represents the user data stored in the data header's respective data segment;

storing a second code comprises a data check code sub-parity sector code into a parity header, the second code representing a set of parity data stored in a parity sector; and,

storing a third code a data check code sub-sector into the parity header, the third code representing a value obtained by encoding the first code and the second code, the third code being used to check the consistency of the user data and the parity data without performing respective operations on each byte of data in both the data sectors and the parity sector;

determining if the user data is corrupt, the determining comprising: reading the parity data and the user data; and for each respective data segment: (a) providing a calculated first code based on the user data stored in the respective data segment; (b) comparing the calculated first code to the first code to determine if the stored user data is corrupt; (c) if the user data is corrupt, regenerating the corrupt user data using certain other data sectors of the data sectors and the parity data; and, (d) storing the regenerated user data in the respective data segment, the regenerated user data replacing the corrupt user data; and

for each respective data sector: (a) if the user data in the respective data sector is corrupt, generating a new first code from the regenerated user data; and, (b) storing the new first code to the respective data sector's respective data header, the new first code replacing the first code.

29. (Original) A computer program product for use in conjunction with a processor, the computer program product comprising a first computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising a data consistency checking procedure to determine inconsistencies between parity data and user data stored in a stripe of data distributed across a plurality of data storage devices in a data storage system, the parity data being stored in a parity sector, the parity sector including a parity header, the user data being stored in one or more data sectors, each respective data sector of the data sectors including a data header, the data consistency checking procedure including instructions for:

for each data header, storing a respective first code comprises a data check code sub-data sector code in the data header that represents the user data stored in the data header's respective data segment;

storing a second code comprises a data check code sub-parity sector code into a parity header, the second code representing a set of parity data stored in a parity sector; and,

storing a third code a data check code sub-sector into the parity header, the third code representing a value obtained by encoding the first code and the second code, the third code being used to check the consistency of the user data and the parity data without performing respective operations on each byte of data in both the data sectors and the parity sector;

determining if the user data is corrupt, the determining comprising: reading the parity data and the user data; and for each respective data segment: (a) providing a calculated first code based on the user data stored in the respective data segment; (b) comparing the calculated first code to the first code to determine if the stored user data is corrupt; (c) if the user data is corrupt, regenerating the corrupt user data using certain other data sectors of the data sectors and the parity data; and, (d) storing the regenerated user data in the respective data segment, the regenerated user data replacing the corrupt user data; and

for each respective data sector: (a) if the user data in the respective data sector is corrupt, generating a new first code from the regenerated user data; and, (b) storing the new first code to the respective data sector's respective data header, the new first code replacing the first code.